# PRINCIPLES OF SECURE SOFTWARE DESIGN

Introductory Lecture

# Factology

- MTAT.03.246, „Principles of Secure Software Design”
- 3 EAP
- Lecturer: Margus Freudenthal
  - Contact: margus@cyber.ee
- Ends with exam
- Mostly lectures, will try to do seminar-like classes where we discuss some topic
- No homework, but there is some mandatory reading
- Literature
  - Main textbook: „Security Engineering” by Ross Anderson
  - Various papers, links will appear on the course home page

# Companion Course

- MTAT.03.247, „Principles of Secure Software Design: Project Work”
- 3 EAP
- Security analysis of a reasonably complex system
- Teams of 2 or 3
- Results in report and presentation

# About Me

- MSc at TTU („Personal Security Environments”), PhD student at TU
- Worked on
  - Digital notary service
  - Time-stamping service
  - Standards
    - ISO time-stamping standard
    - Estonian digital signature standard
  - X-Road security infrastructure
  - Security audits, security testing

# What This Course Is About?

- Security implications of high-level design decisions
- Security issues arising from interaction of complex systems
- Protecting against right threats
- Security economics, both from defender's and attacker's side

# What This Course Is About? (2)

- What are typical security measures in various domains? What are typical attacks?
- Focus is on domain-specific threats instead of generic ones (e.g., CSS, injection, buffer exploits etc.)

# This Course Will Not Cover

- Cryptography
  - In some later topics, basic knowledge about main cryptographic primitives is assumed
- Low-level vulnerabilities (e.g., buffer overflows), how to avoid them

# Topics

- Designing secure software
- Designing security protocols
- Human factors and security
- Authentication, biometrics
- Economics of software security
- Development process for secure software

# Topics: Examples

- Multilevel security, multilateral security
- Banking and bookkeeping
- PKI and digital signatures
- Online games
- X-Road
- e-elections

# What Is the System?

- A component, such as smartcard, a PC or piece of software

# What Is the System?

- A component, such as smartcard, a PC or piece of software
- Above + operating system, network and other infrastructure

# What Is the System?

- A component, such as smartcard, a PC or piece of software
- Above + operating system, network and other infrastructure
- Above + applications

# What Is the System?

- A component, such as smartcard, a PC or piece of software
- Above + operating system, network and other infrastructure
- Above + applications
- Above + IT staff

# What Is the System?

- A component, such as smartcard, a PC or piece of software
- Above + operating system, network and other infrastructure
- Above + applications
- Above + IT staff
- Above + internal users and management

# What Is the System?

- A component, such as smartcard, a PC or piece of software
- Above + operating system, network and other infrastructure
- Above + applications
- Above + IT staff
- Above + internal users and management
- Above + customers and external users

# What Is the System?

- A component, such as smartcard, a PC or piece of software
- Above + operating system, network and other infrastructure
- Above + applications
- Above + IT staff
- Above + internal users and management
- Above + customers and external users
- Above + environment

# What Is the System?

- A component, such as smartcard, a PC or piece of software
- Above + operating system, network and other infrastructure
- Above + applications
- Above + IT staff
- Above + internal users and management
- Above + customers and external users
- Above + environment

**Everything**

# Next, Some Examples...

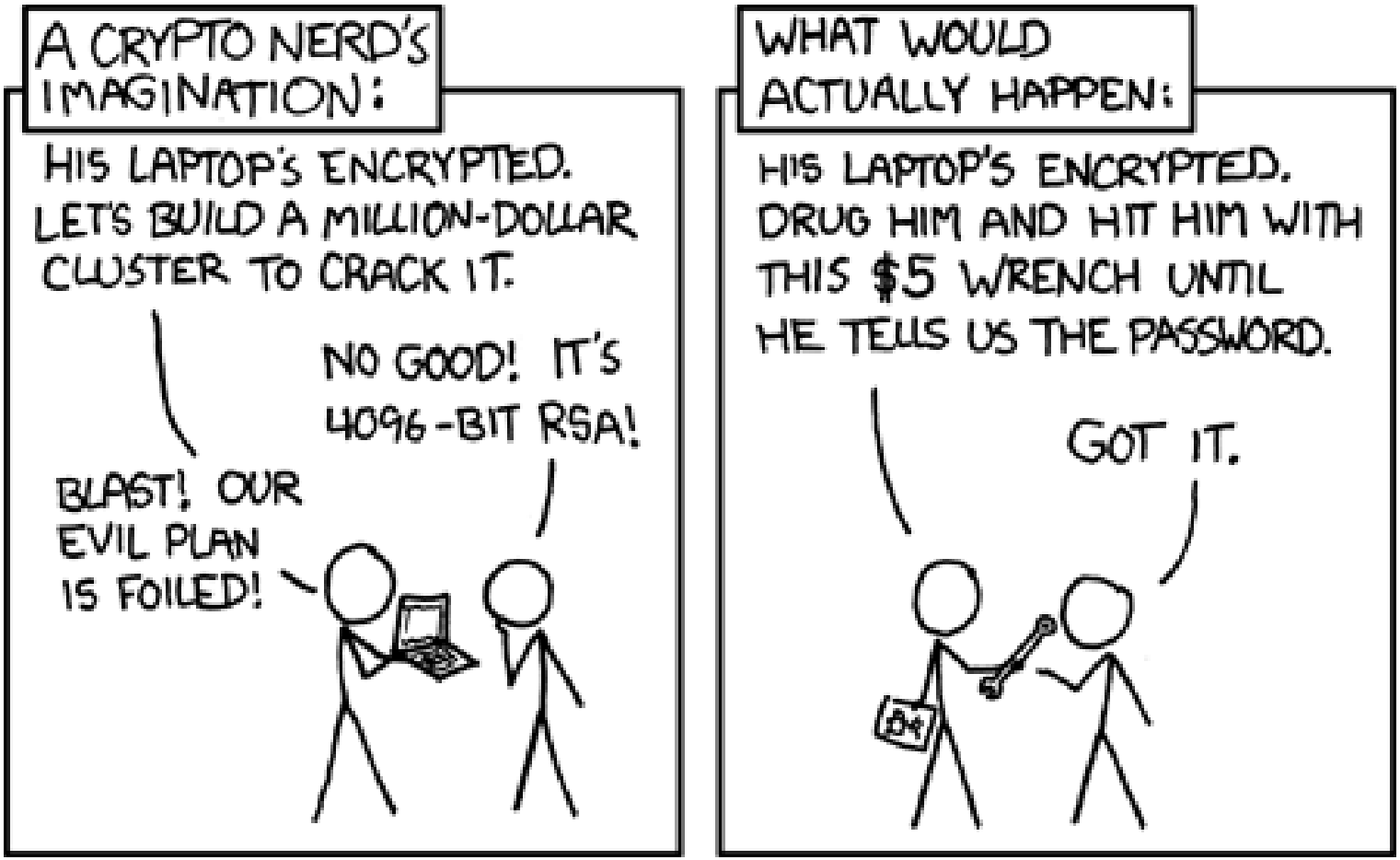
# How to Crack Encrypted Message?

- Acquire massive amount of computing power and brute-force all the possible values of the encryption key?
  - The cryptographer's dream scenario
  - Although, what about the case where the key is easily-guessable password?

# How to Crack Encrypted Message?

- Acquire massive amount of computing power and brute-force all the possible values of the encryption key?
  - The cryptographer's dream scenario
  - Although, what about the case where the key is easily-guessable password?
- Coerce extremely brilliant (albeit eccentric and socially inept) mathematician to crack the encryption algorithm
  - The Hollywood version
  - Bonus points if the mathematician is a teenager

# The XKCD Version



# How to Crack Encrypted Message?

- Or, better yet, why not just ask for password?
  - ... or the message itself

# How to Crack Encrypted Message?

- Or, better yet, why not just ask for password?
  - ... or the message itself
- How about accessing the computer and installing keylogger or trojanized version of the message viewer?
  - Chances are, the user already has some remote-controlled malware installed
  - Also, maybe the decrypted message can be read from computer's memory or hard disk?

# A Shop Example

# A Security Trade-Off

- Supermarkets have considerably higher risk of theft than smaller, over the counter stores
- However...
  - Economy of scale
  - Less personnel
- It gets better
  - Self-service checkout
  - Customers can weigh items themselves

# Point to Note

- Organization's goal is usually not to secure stuff
- Instead:
  - Make profit selling books
  - Provide better public service to citizens
- Sometimes, less security can better fulfil organization's goals

# Example: Traffic Lights

- Johannesburg installed networked traffic lights
- Connects via GSM network
- Thieves steal SIM cards to make phone calls

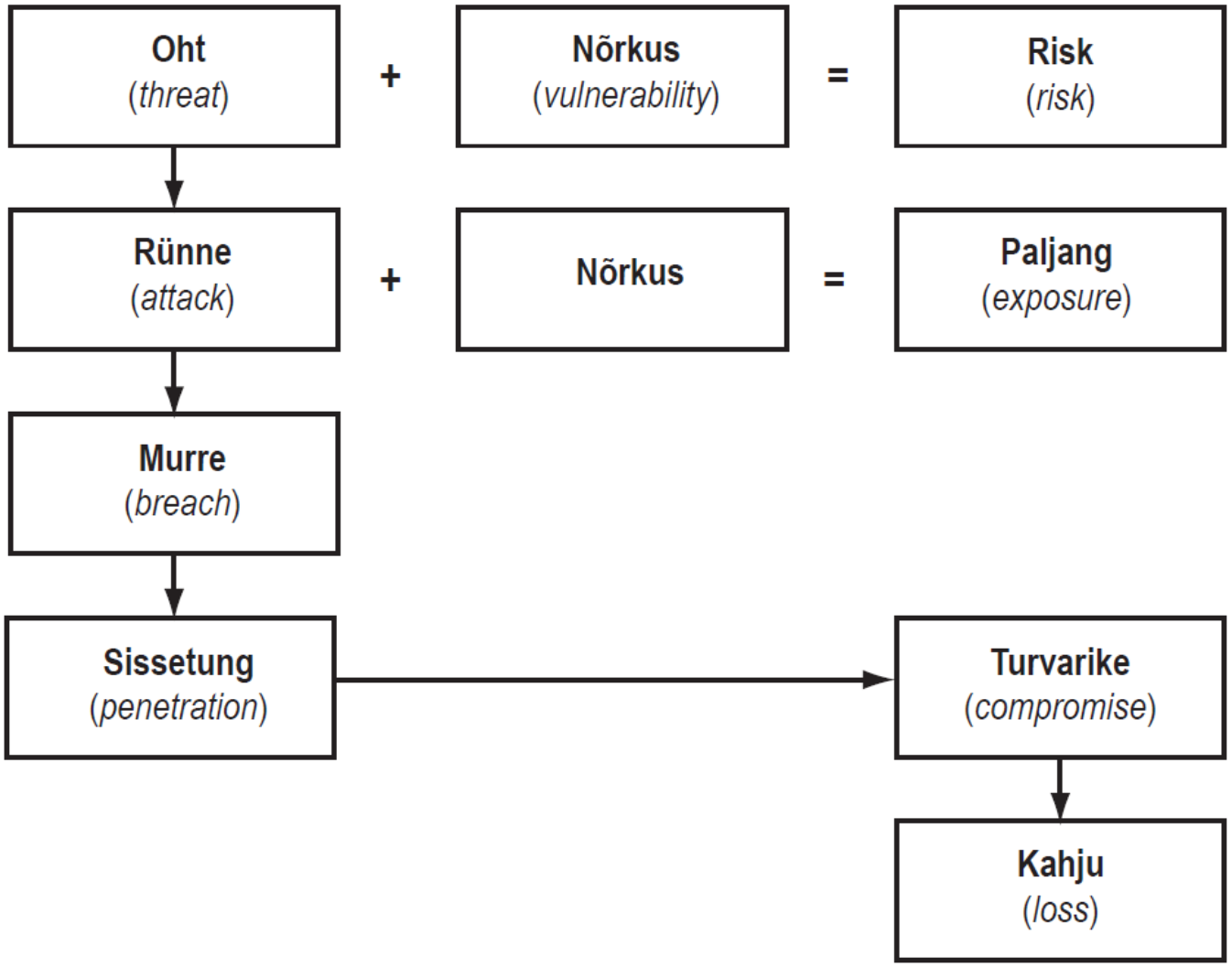
# Example: the *ls* Command

- Suppose the Unix *ls* command contains buffer overflow when processing input parameters
  - The ls command lists files in the directory
- Is this a security vulnerability?

# Example: the *ls* Command (2)

- For ordinary command-line usage, this is a reliability issue
- It becomes security issue if *ls* is used in a boundary between two security zones
  - For example, in anonymous FTP server
    - FTP server separates untrusted Internet from trusted file system
  - The *ls* command runs under user X, but parameters come from a less trusted entity

# Some Terminology



# Some Terminology (2)

- Trusted system – a component or system whose failure will break the security policy
- Trustworthy system – component or system that will not fail

# Some Terminology (3)

- Security policy – succinct statements of system's protection strategy
  - e.g. „each credit must be matched by an equal and opposite debit, and all transactions over $1000 must be authorized by two managers”
- Security target – more detailed specification which describes how the security policy is implemented

# The Three Parts of Security

- Prevention – bad things do not happen
- Detection – if bad things happen, we'll know about it
- Response – if attack is detected, do something about it

# Example: A Building

- Prevention: building 5m high solid steel wall with electricity running through it
- Detection: installing security cameras
- Response: catching and suing criminals, insuring assets

# Some Notes

- Prevention is often impossible or unreasonable
  - In real life, this is replaced by deterrence, inconveniencing, etc.
- Detection is usually meaningful if some kind of recovery is attempted

# Thank You